Databases -2- Practical Labs

3rd Year





Lab -2- PL/SQL Program Structure

Introduced By:

Eng. Khaled Kondakji *Eng. Ibrahim Ayoub*

Eng. Aisha Jennyat

Supervised By:

Dr. Eng. Asmaa Shaar

PL/SQL Program Structure

العبارات الشرطية (IF Statements):

لفهم آلية كتابة العبارات الشرطية سنقوم بتنفيذ المثال التالي:

```
DECLARE
    V_LETTER CHAR(1) := '&sv_letter'; --take input from user

BEGIN
    IF (V_LETTER >= 'A' AND V_LETTER <= 'Z') OR  --start IF condition
       (V_LETTER >= 'a' AND V_LETTER <= 'z') THEN -- if condition is true the
next statements will be executed
        DBMS_OUTPUT.PUT_LINE('This is a letter '||V_LETTER);
    ELSE -- if condition is false the next statements will be executed
        DBMS_OUTPUT.PUT_LINE('This is not a letter ');
        IF V_LETTER BETWEEN '0' AND '9' THEN
            DBMS_OUTPUT.PUT_LINE('This is a number '||V_LETTER);
        ELSE
            DBMS_OUTPUT.PUT_LINE('This is not a number ');
        END IF;
    END IF;
END;
```

يعمل البرنامج السابق وفق المراحل التالية:

1- يتم إدخال قيمة من قبل المستخدم ومن ثم تتحقق عبارة IF الخارجية فيما إن كانت القيمة المدخلة هي محرف أم لا، حيث يتم تنفيذ التعليمات التي تأتي بعد THEN عند تحقق الشرط وإلا يتم تنفيذ التعليمات الموجودة بعد العبارة ELSE.

2- عند تنفيذ التعليمات الموجودة بعد عبارة ELSE يتم فحص القيمة المدخلة من قبل المستخدم فيما إن كانت رقماً أم لا ويتم تنفيذ التعليمة المناسبة بحسب القيمة المدخلة.

للتنفيذ: قم بتنفيذ البرنامج السابق من أجل القيم التالية (M, r, 9, @) ولاحظ خرج البرنامج السابق.

كمثال عند إدخال القيمة (m) مثلاً تكون نتيجة تنفيذ البرنامج كالتالي:

Input Parameters

Enter value of sv_letter:

m

ومن ثم يكون الخرج كما يلي:

```
Enter value for sv_letter:
m

old   2:     V_LETTER CHAR(1) := '&sv_letter';

new   2:     V_LETTER CHAR(1) := 'm';

This is a letter m
```

تمرين: قم بإضافة شرط جديد على بنية IF الداخلية باستخدام البنية ELSIF للتحقق من القيمة فيما إذا كانت إشارة (@) وطباعة العبارة الموافقة ومن ثم تعديل عبارة الطباعة ضمن ELSE الداخلية لتطبع أن القيمة المدخلة ليست رقم أو إشارة (@).

استخدام التعليمة (CASE):

تساعد هذه التعليمة على اختيار أي مجموعة من التعليمات سيتم تنفيذها بناءً على نتيجة تعبير معين. كما في المثال التالي:

```
-- this statement must be executed before run the code:
ALTER SESSION SET CURRENT_SCHEMA = HR;
---------------------------------------------------------------
-- Now we're executing on HR Schema.
DECLARE
    V_EMPID NUMBER;
    V_NAME  VARCHAR2(50);
    V_MID   VARCHAR2(50); -- manager ID
BEGIN
    SELECT
        EMPLOYEE_ID, FIRST_NAME,
        CASE
            WHEN MANAGER_ID IS NULL THEN 'The BOSS'
            ELSE
                TO_CHAR(MANAGER_ID) -- if not the boss or manager id is not null

        END
        INTO V_EMPID, V_NAME, V_MID
    FROM EMPLOYEES
    WHERE EMPLOYEE_ID = 100;
    DBMS_OUTPUT.PUT_LINE('Employee id: ' ||V_EMPID);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' ||V_NAME);
    DBMS_OUTPUT.PUT_LINE('Manager ID: ' ||V_MID);
END;
```

ويكون خرج البرنامج السابق من أجل الموظف ذو المعرف (100):

```
Employee id: 100
Employee Name: Steven
Manager ID: The BOSS

PL/SQL procedure successfully completed.

Commit complete.
```

أما من أجل الموظف ذو المعرف (200):

```
Employee id: 200
Employee Name: Jennifer
Manager ID: 101

PL/SQL procedure successfully completed.

Commit complete.
```

تمرين: قم بتعديل البرنامج السابق بحيث يقوم بالاستعلام عن أسماء الموظفين وأرقامهم، ثم قم بتطبيق تعليمة CASE على حقل معرف القسم بحيث يطبع ما يلي:

- إذا كان رقم القسم يساوي NULL يطبع عبارة الموظف الحالي لا يتبع لقسم كالموظف رقم (178).
- إذا كان رقم القسم يساوي 90 يطبع عبارة الموظف الحالي يتبع لقسم المدير.
- في حالة عدم تحقق أي من الشروط السابقة نطبع رقم القسم فقط.

استخدام تابع (COALESCE):

يستخدم هذا التابع لاختبار مجموعة تعابير فيما إذا كانت نتيجتها NULL أم لا وتعيد أول نتيجة أول تعبير لا تساوي القيمة NULL، وفيما يلي سنقوم بتعديل المثال السابق بحيث يستخدم هذا التابع بدلاً من استخدام تعليمة CASE.

```
-- this statement must be executed before run the code:
ALTER SESSION SET CURRENT_SCHEMA = HR;
---------------------------------------------------------------
-- Now we're executing on HR Schema.
DECLARE
    V_EMPID NUMBER;
    V_NAME  VARCHAR2(50);
    V_MID   VARCHAR2(50); -- manager ID
BEGIN
    SELECT
        EMPLOYEE_ID, FIRST_NAME,
        COALESCE(TO_CHAR(MANAGER_ID), 'The BOSS')
        INTO V_EMPID, V_NAME, V_MID
    FROM EMPLOYEES
    WHERE EMPLOYEE_ID = 100;
    DBMS_OUTPUT.PUT_LINE('Employee id: ' ||V_EMPID);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' ||V_NAME);
    DBMS_OUTPUT.PUT_LINE('Manager ID: ' ||V_MID);
END;
```

ويكون خرج البرنامج مشابهاً للمثال السابق.